META-META Club

WORKBOOK · MODULE 3 OF 5

Three.js fundamentals

Browser Worlds. Scene, camera, renderer, glTF, light and walk controls.

Pro studios | 3 lessons | ~3.3 h | 5-question practice quiz | Notes pages

IN THIS MODULE

Three.js fundamentals

Scene, camera, renderer, glTF, light and walk controls.

#	LESSON	TYPE	TIME	DONE
3.1	Scene, camera, renderer	LAB	60 min	☐
3.2	Load glTF and light it	LAB	60 min	☐
3.3	Walk controls	LAB	75 min	☐

How to use this workbook

- Work through each lesson on metametaclub.com/learn, or listen to its narration, then use the page here to take notes.
- Tick the steps as you go, and write your answers under **Check yourself** before you look anything up.
- Do the practice quiz at the end, then mark it with the answer key on the last pages.
- Keep every **Ship it** piece. Together they become what you show.

Before you start

What do you already know about this, and what do you want to be able to make by the end?

LESSON 3.1 · LAB · 60 MIN

Scene, camera, renderer

Build the smallest correct Three.js page: a scene, a camera, a renderer, one object and a render loop that handles window resizing. Every browser world in this course starts from this file.

Key points

The three parts

- **Scene.** The container for everything you draw: meshes, lights, helpers.
- **Camera.** Where you look from. `PerspectiveCamera` takes a field of view, an aspect ratio and near and far clipping distances.
- **Renderer.** Draws the scene from the camera into a canvas using WebGL.

A **mesh** is geometry (the shape) plus a material (how the surface responds to light). The **render loop** redraws every frame.

Steps checklist

- [ ] Make a folder with one file, `index.html`, and paste the code below.
- [ ] Serve the folder: `npx serve .` and open the address it prints. Modules don't load from `file://`.
- [ ] Resize the window. The cube should stay square and sharp.
- [ ] Change the colour, the size and the camera position. Break something on purpose and read the console error.

Units are metres by convention. The camera sits at eye height, 1.6 m.

```
This lesson has code samples. They are printed in full in the lab handout (browser-worlds-lab-3-1.pdf)
and in the lesson on metametaclub.com/learn.
```

SHIP IT

Your page, served locally, with a coloured object of your choice and a screenshot posted to your notes.

- [ ] Shipped. Link or file: _____

Check yourself

1. Why does the page need a local server?

2. What two things must you update when the window resizes?

3. Why cap the pixel ratio at 2?

LESSON 3.2 · LAB · 60 MIN

Load glTF and light it

Load a real 3D model in glTF format, light it with a sky and a sun, turn on shadows and set colour so materials look right. This is how most of your world will arrive: modelled in Blender, exported as `.glb`, loaded in the browser.

Key points

Why glTF

glTF is the open standard for 3D on the web. A `.glb` file packs geometry, materials, textures and animation into one download.

Light in two layers

- **HemisphereLight** fakes sky light: a sky colour from above, a ground colour from below. Cheap and soft.
- **DirectionalLight** is the sun: parallel rays, sharp shadows. Shadows cost performance, so give the shadow camera tight bounds around your scene.

Colour

Three.js outputs sRGB by default and GLTFLoader sets texture colour spaces for you. Add tone mapping so bright skies don't clip to white.

Steps checklist

- [ ] Start from your Scene, camera, renderer page.
- [ ] Put a `model.glb` next to `index.html`. Your own export, or a sample from the Khronos glTF sample assets.
- [ ] Replace the cube and light with the code below.
- [ ] Move the sun. Watch shadows swing. Tune `sun.shadow.camera` until shadows are crisp, not blocky.

Top-level `await` works inside a module script. Add a ground plane with `receiveShadow = true` so the model sits on something.

```
This lesson has code samples. They are printed in full in the lab handout (browser-worlds-lab-3-2.pdf)
and in the lesson on metametaclub.com/learn.
```

SHIP IT

Your model loaded, lit and casting a shadow, with a screenshot and the file size of the `.glb` noted.

- [ ] Shipped. Link or file: _____

Check yourself

1. What does each of the two lights do?

2. Why set tight shadow camera bounds?

3. What does tone mapping fix?

LESSON 3.3 · LAB · 75 MIN

Walk controls

Add first-person walking: click to capture the mouse, move it to look, WASD to walk. Movement uses frame time so it feels the same on every device.

Key points

Pointer lock

Browsers won't let a page grab the mouse without a click. The Pointer Lock API hides the cursor and reports raw movement, which `PointerLockControls` turns into looking around.

Frame time with Timer

Since r183, `THREE.Clock` is deprecated. Use `THREE.Timer`: call `update()` once per frame, then `getDelta()` returns the seconds since the last frame.

Make it feel good

- Show a short "Click to walk · WASD · Esc to release" hint and hide it on the `lock` event.
- Diagonal movement is faster than straight. Normalise the input if it bothers you.
- Phones have no pointer lock. Plan a touch fallback: tap-to-move or an on-screen stick.

Steps checklist

- [ ] Start from your lit glTF page.
- [ ] Add the code below after the scene is set up.
- [ ] Set walking speed to about 1.4 m/s. Try 4 m/s for big sites.
- [ ] Add bounds so people can't walk off the world.
- [ ] Test on a laptop trackpad and a mouse.

`moveForward` and `moveRight` move along the ground plane, so looking up doesn't make you fly. Keep `camera.position.y` at 1.6.

```
This lesson has code samples. They are printed in full in the lab handout (browser-worlds-lab-3-3.pdf)
and in the lesson on metametaclub.com/learn.
```

SHIP IT

Your scene, walkable, with bounds and an on-screen hint, plus a 10-second clip.

- [ ] Shipped. Link or file: _____

Check yourself

1. Why does the page need a click before pointer lock works?

2. What would go wrong without `dt`?

3. Why use Timer instead of Clock in r185?

PRACTICE QUIZ

Check your module

Circle one answer for each question. The answer key is on the next page.

1. Why use renderer.setAnimationLoop instead of your own requestAnimationFrame loop?

(A) It renders at a higher resolution

(B) It keeps working when you enter a WebXR session

(C) It loads models faster

2. What must you update when the browser window resizes?

(A) The scene background

(B) The camera aspect (then updateProjectionMatrix) and the renderer size

(C) Every material

3. Which light gives soft sky light from above and ground colour from below?

(A) DirectionalLight

(B) PointLight

(C) HemisphereLight

4. In Three.js r185, what replaces the deprecated Clock for frame time?

(A) THREE.Timer, with update() each frame and getDelta()

(B) Date.now()

(C) THREE.Stopwatch

5. Why won't an ES module page load when you double-click the HTML file?

(A) Three.js needs a licence key

(B) Browsers block module imports from file:// URLs, so you need a local server

(C) The GPU is turned off

ANSWER KEY

Answers and why

1 · B It keeps working when you enter a WebXR session

In XR the headset drives the frame loop. setAnimationLoop hands over to it automatically.

2 · B The camera aspect (then updateProjectionMatrix) and the renderer size

Without both, the image stretches or renders at the wrong size.

3 · C HemisphereLight

HemisphereLight blends a sky colour and a ground colour. It is cheap fill light.

4 · A THREE.Timer, with update() each frame and getDelta()

Timer was added to fix Clock's pitfalls, and Clock is deprecated since r183.

5 · B Browsers block module imports from file:// URLs, so you need a local server

Module scripts follow CORS rules that file:// can't satisfy. Serve the folder with npx serve or similar.

NOTES

Build log and notes

SESSION	WHAT I MADE OR LEARNED	NEXT STEP